














SEARCH RESULTS


You searched for: ((database) AND backup)

You refined by:

Subscribed Content: Draft , IBM 

Content Type: Conferences , Journals , Books , Standards , Educational Courses 

Subject: Computing & Processing (Hardware/Software) , Components, Circuits, Devices & Systems ,
Signal Processing & Analysis , Communication, Networking & Broadcasting , Power, Energy, &
Engineers (Math, Science & Engineering) , Engineering Profession 

Results per Page 

Storing and using objects in a relational database

Reinwald, B.; Lehman, T. J.; Pirahesh, H.; Gottemukkala, V.;

IBM Systems Journal

Volume: 35 , Issue: 2

Digital Object Identifier: 10.1147/sj.352.0172

Publication Year: 1996 , Page(s): 172 - 191

IBM JOURNALS

In today's heterogeneous development environments, application programmers have the responsibility to store those data in different types of stores. That means relational data will be stored in RDBMSs (relational database management systems), OODBMSs (object-oriented database management systems), SOM (System Object Model) objects in OpenDoc™ or OLE™ (Object Linking and Embedding) compound documents in document files. In a multiple server systems with different query languages as well as large amounts of heterogeneous data resident cache), an RDBMS extender that provides the ability to store objects created in external type systems corident with existing relational or other heterogeneous data. Using SMRC, applications can store and retrieve objects (in any language), and invoke methods on the objects, without requiring any modifications to the original objects. This paper describes how to participate in all the characteristic features of the underlying relational database, e.g., transactions, backup and recovery, and so on. It is based on top of IBM's DB2® Common Server for AIX® relational database system and heavily exploits the DB2® user-defined functions (UDFs), and large objects (LOBs) technology. In this paper, the C++ type system is used as a sample approach, i.e., storing C++ objects in relational databases. Similar efforts are required for SOM or OLE